Journal of Computer and Knowledge Engineering, Vol. 6, No. 2, 2023. (15-24)

Ferdowsi University of Mashhad

Journal of Computer and Knowledge Engineering

https://cke.um.ac.ir

Information and Communication Technology Association of Iran

An Efficient Ramp Secret Sharing Scheme Based on Zigzag-Decodable Codes*

Research Article

Saeideh Kabirirad[1], Sorour Sheidani[2], Ziba Eslami [3]

DOI: 10.22067/cke.2023.83418.1094

Abstract: Secret sharing schemes are ideally suited to save highly sensitive information in distributed systems. On the other hand, Zigzag-Decodable (ZD) codes are employed in wireless distributed platforms for encoding data using only bit-wise shift and XOR operations. Recently, Vandermonde-based ZD codes have been utilized in secret sharing schemes to achieve high computational efficiency such that sharing and recovering of secrets can be realized by lightweight operations. However, the storage overhead of using these ZD codes remains a problem which is addressed in the present paper. Here, a ramp secret sharing scheme is proposed based on an efficient ZD code with less storage overhead in comparison with existing literature. The novelty of the proposed scheme lies in the careful selection of the number of positions to shift the bits of the secret such that security and zigzag decodability are guaranteed simultaneously. In addition to prove gaining these features, we show that the scheme improves speed of recovery.

Keywords: Boolean operation, Efficiency, Ramp secret sharing scheme, Zigzag decodable codes.

1. Introduction

Security is a vital necessity in distributed systems and cloud environments. With the rise in cloud computing environments and Internet of things systems, secret sharing (SS) schemes have become increasingly important cryptographic primitives. In an SS scheme, a secret is distributed to some shares such that any qualified subset of shareholders can reconstruct the secret while no unqualified subset can gain any information about it. SS schemes are used as the key element of numerous security protocols, most notably in distributed storage systems, threshold cryptography and secure multi-party computation. However, in many of these

applications, lightweight schemes are a requirement. In this context, there have been efforts to present the schemes that can be implemented using only Boolean operations, namely Shift and XOR.

(t, n)-threshold SS schemes constitute an important type of secret sharing schemes. During these schemes, a secret is distributed among n participants in a way that any t or more participants be able to reconstruct the secret [1]. A large number of methods have been applied to improve the efficiency of threshold schemes in which using the boolean operations in the schemes establish a considerable part of it [2].

In [3], Shiina et al. presented a (t, n)-threshold SS scheme to improve Shamir's threshold SS scheme. In spite of the improvement in terms of computational time, their method imposes a large storage overhead for the shares. Kurihara et al. [4] presented a $(3, n)$-threshold SS scheme using XOR that was *ideal* (i.e., the domain of the shares and the secret are equal). Then, they generalized their method in [5] to any arbitrary threshold value by presenting an ideal (t, n)-threshold SS scheme using boolean operations. They also extended the (t, n)-threshold SS and provided the concept of a (t, w, n)-Ramp Secret Sharing (RSS) scheme [6] where w determines a boundary for the minimum number of participants who have to form a coalition to achieve some information about the secret [7].

In Kurihara et al.'s schemes, the shares are computed by applying XOR operation to the secret pieces and the sequences of random bits. In their schemes, after collecting the required number of shares, they are saved in a vector. Then, the secret can be reconstructed through multiplying this vector and a matrix calculated according to Gaussian elimination. Although Gaussian elimination imposes high computational complexity in practice, for some parameters, it is experimentally shown that this elimination is faster than Shamir's scheme. [8] discusses the subject of the scheme of Kurihara et al.'s using the properties of circular matrices. The authors achieve a new security analysis for these secret sharing schemes. Afterward, Wang and Desmedt [9] presented a (t, n)-threshold scheme which needed just XOR and cyclic shift operations. Additionally, Chen et al. [10] who recommended a boolean-based (t, n)-threshold scheme, claimed that it is more efficient than Kurihara et. al's and also Wang and Desmedt's schemes. However, their scheme has a limitation and cannot be used in general case, i.e. it works

* Manuscript received: 2023 July 14, Revised, 2023 August 20 , Accepted, 2023 September 30.

[1] Corresponding author. Assistant professor, Department of Computer Science, Birjand University of Technology, Birjand, Iran.
Email: kabiri@birjandut.ac.ir.
[2] Ph.D., Department of Data and Computer Sciences, Shahid Beheshti University, G.C., Tehran, Iran.
[3] Associate professor, Department of Data and Computer Sciences, Shahid Beheshti University, G.C., Tehran, Iran.

only for $n = t, t+1$. Shima et al. [11] suggested a way to fix this problem and then extended the improved method to a hierarchical scheme. In another work [12], a (n, n) −threshold secret sharing scheme based on binary trees and XOR operation has been proposed. Some studies applied boolean operations to optimize common SS schemes such as Shamir's method [13].

Boolean-based methods have also been considered for large secrets such as images [14]–[16]. The threshold schemes employed by lightweight operations such as shift and XOR benefit from high computational efficiency, lossless image recovery, ability of multiple images sharing, supporting any image formats, fast recovery, etc [17]- [20].

Lately, zigzag-decodable (ZD) erasure codes have been employed in conjunction with ramp secret sharing schemes. ZD codes are the first XOR-based Maximum Distance Separable (MDS) codes proposed for distributed storage systems [21] that can correct node failures. In ZD codes, both encoding and decoding processes are carried out exclusively by the operators of bitwise shift and XOR, without the need for finite field multiplication. Also, it is possible to perform decoding process easily via physical layer network coding [22]. These advantages imply that in a wide variety of applications where efficiency is important (such as big-data storage, wireless distributed storage systems and resource-constrained devices), ZD-based schemes can be conveniently employed [23], [24].

However, this efficiency is at the expense of some storage overhead. In [25], Gong et al. proposed a (t, w, n)-RSS scheme which has been adapted from ZD codes based on Vandermonde matrix, abbreviated as VZD-RSS. Their scheme inherited the features of ZD codes, i.e., it has less computational complexity compared to the schemes in the literature and also has some storage overhead.

In [21], the authors presented a sufficient condition, so-called "increasing difference" property, for enabling zigzag decodability . Based on this condition, two constructions based on the Vandermonde and the Hankel matrices were proposed [26]. Afterwards, Dai et al. [27] presented another ZD code using circular matrices with less storage overhead. Their method does not provide the sufficient condition and they demonstrate a new proof for constructing feasible ZD code.

1.1. Our Contributions

The main contribution of the present paper is proposing a secure (t, n)-RSS scheme based on ZD codes with less storage overhead than VZD-RSS. That is, we introduce a new generator matrix that reduces size of shares compared to the Vandermonde-based schemes while maintaining security. Here, our contributions in comparison with the existing literature are briefly listed:

1. Our proposed scheme supports arbitrary values of n and t, has low computational complexity, and provides zigzag decodability.
2. Our proposed scheme reduces storage overhead by almost half compared to Vandermonde-based scheme. The overhead can be neglected for large secrets or cases where n and t are close.
3. Our proposed scheme is proven to achieve zigzag decodability property according to new conditions.
4. Our proposed scheme satisfies security requirements.

1.2. Paper Organization

The organization of the rest of the paper is as follows: Section II reviews some definitions of RSS schemes. Section III introduces ZD codes. The proposed scheme is presented in Section IV. Section V describes security analysis and conditions of zigzag decodability. In Section VI, the efficiency of the proposed scheme is analyzed and compared with boolean-based methods in the literature. Conclusions are presented in Section VII and finally, we provide the details of parts of our proofs in the Appendix section.

2. C Preliminaries

In this section, we review required definitions and provide necessary notations for ramp secret sharing (RSS) schemes.

Let X and Y be two jointly distributed random variables. Let $H(X)$ denote the $Shannon\ entropy$ of X and let $H(X|Y)$ be the $conditional\ entropy$ of X given Y.

According to Shannon entropy, $H(X|Y) = 0$ indicates that X is a deterministic function of Y. However, $H(X|Y = y) = H(X)$ indicates that in case $\{Y = y\}$, no information about X is leaked.

Definition: Assume t, w and n are integers where $0 < w \leq t \leq n$. A (t, w, n)-*RSS scheme* distributes a secret message K among n participants such that two conditions hold:

1. *Decodability.* Any subset A of t or more participants, can uniquely recover K, i.e., $H(K|A) = 0$.
2. *Secrecy.* Any set A' of at most $(t - w)$ participants, obtains no information about K, i.e., $H(K|A') = H(K)$.

By definition, a $(t, 1, n)$-RSS scheme is a (t, n)-threshold secret sharing scheme. In fact, RSS schemes are solutions to reduce the size of shares while losing secrecy to some extent.

Definition: A (t, w, n)-ramp secret sharing scheme is *linear* if for any subset of participants A that $|A| = r$ and $t - w < r < t$, we have $H(K|A) = \frac{w-r}{w} H(K)$.

It means that, after pooling $(t - w)$ shares, every further share reveals $\frac{1}{w}$ bits of information about the secret K [27].

3. Review of ZD Codes

The encoding and decoding processes of ZD codes are based solely on boolean operations, including shift and XOR.

Zigzag decodability is the ability of recovering the original data by zigzag decoding [28]. In this section, the coding and decoding processes of ZD codes are reviewed in general.

3.1. Coding

Given a message K with length $\lambda = tL$ bits, we split it into t

pieces $K_1, K_2, \dots, K_t$.

The bit-length of each piece of message K_i is L bits. Furthermore, polynomial representation of K_i is:

$$K_i(z) = K_{i,0} + K_{i,1}\, z + \cdots + K_{i,(L-1)}\, z^{L-1} \quad (1)$$

where $K_{i,j}$ is an element in $GF(2)$.

By linear combination of the t pieces of the message, n encoded packets $C_1(z), C_2\,(z), \dots, C_n(z)$ are generated.

Each $C_i(Z), i = 1,2, \dots, n$ is $L' = L + l$ bits long, where l denotes the storage overhead, i.e., the encoded packets are l bits longer than the pieces of message. Then, the polynomial representation of $C_i(z)$ is given by:

$$C_i(z) = C_{i,0} + C_{i,1}\, z + \cdots + C_{i,(L'-1)}\, z^{L'-1} \quad (2)$$

Each $C_i(z)$ is generated in two phases: 1) shifting pieces of message and 2) adding them. Hence, the i-th encoded packet is obtained as follows:

$$C_i(z) = z^{e_{i,1}}\, K_1 + z^{e_{i,2}}\, K_2 + \cdots + z^{e_{i,t}}\, K_t \quad (3)$$

where $e_{i,j} \in Z$. Note that multiplying by z^j means shifting by j positions while add operation (performed in $GF(2)$) means XOR.

According to (3), storage overhead of each encoded packet will be $l = max_{i,j}\{e_{i,j}\}$. Considering the source and encoded data, the corresponding matrix form is:

$$C(z) = G(z) \times K(z) \quad (4)$$

where $C(z)$ is a vector of length n and its i-th element is $C_i(z)$. Additionally, $K(z)$ is a t-dimensional vector containing pieces of message.

$G(z)$ is called the generator matrix and is an $n \times t$ matrix with $z^{e_{i,j}}$ as (i,j)-th element. Note that matrix $G(z)$ is t-reliable, this implies that any $t \times t$ submatrix of $G(z)$ can be used to recover the pieces of message.

So far, there are some suggestions for matrix $G(z)$ in the literature, such as Vandermonde, Hankel, etc. In VZD-RSS [25], choosing Vandermonde matrix has fulfilled the security requirements and the storage overhead equals $(n-1)(t-1)$. In Section IV, we propose a generator matrix such that the storage overhead is reduced by half compared to VZD-RSS.

3.2. Zigzag Decoding

Suppose that t arbitrary coded packets are available. We now describe how the source packets are recovered by zigzag decoding algorithm.

First, a $t \times t$ submatrix $M(z) = [z^{g_{i,j}}]$ of $G(z)$ is constructed using the corresponding indices of the available encoded packets. Consider a $t \times t$ integer matrix $E = [e_{i,j}]$ in which its elements are exponents of the corresponding elements in $M(z)$. The main idea of zigzag decoding algorithm is to find an encoded packet that has a bit which can be directly extracted. Such a bit is called an "exposed" bit.

Afterwards, the bit is deduced from other encoded packets. This process is done repeatedly until recovering all source bits. In Figure 1, an example of zigzag decoding procedure with two encoded packets is shown. The computational complexity of zigzag decoding is $O(t^2 L)$.

In the following, we review the details of zigzag decoding algorithm as stated in [21].

Let i be the index of an encoded packet and similarly, j be the index of a source packet. Also, let m and m' be the set of indexes of the encoded packets and set of indexes of unrecovered source packets. The polynomials $\hat{x}_j(z)$ and $y_i(z)$ are the decoded portion of j-th source packet and also the not decoded part of i-th packet.

Furthermore, for a polynomial $f(z)$, consider $\Omega(f(z))$ and $\omega(f(z))$ as the term with the smallest order and the order of that term, respectively.

Zigzag Decoding Algorithm:

Step 1: (Initialization) Let $m' := m$ and $\hat{x}(z) :=\ 0$. Let

$$\eta_j(z) := 1 + z\ + \cdots + z^{L+l-1}, \sim \text{for all } j \in\ m' \quad (5)$$

Step 2: (Searching for an exposed bit) Find an $i^* \in m$ and some $j^* \in\ m$ such that

$$\omega(z^{t_{i^*,j^*}} \eta_{j^*}(z))\ < \omega(z^{i^*,j} \eta_j(z)) \text{ for all } j \in m' \setminus \{j^*\} \quad (6)$$

Step 3: (Updating variables)

1. Let $\hat{x}_{j^*}(z) := \hat{x}_{j^*}(z)\ + \Omega(y_{i^*}(z))$.
2. Let $y_i(z) :=\ y_i(z) -\ z^{i,j^*}\Omega(x_{j^*}(z))$ for all $i \in m$.
3. Remove the term of $\eta_{j^*}(z)$ which has the smallest order. If there is no more term in $\eta_{j^*}(z)$, }delete j^* from m'.

Step 4: If $m' \neq 0$ go to Step 2, else exit and output $\hat{x}_j(z)$ for all $j \in m$.

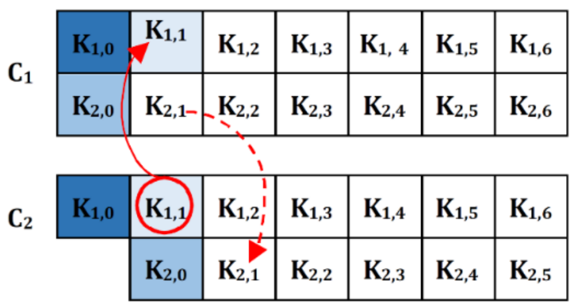


Figure 1. Illustration of zigzag decoding for two coded packets.

4. The Proposed (t,w,n)-RSS Scheme

In this section, we first explain our ZD code that decreases storage overhead in the recovery phase. This is achieved by substituting the ZD code's Vandermonde matrix by another modified matrix. Then, we propose a (t, w, n)-RSS scheme using this ZD code. Our proposed generator matrix G is defined as (7).

$$G = \left(\begin{array}{cccc|cccc} 1 & \dots & z^{t-w-1} & z^{t-w} & z^{t-w+1} & \dots & z^{t-1} \\ 1 & \dots & z^{2(t-w-1)} & z^{2(t-w)} & z^{2(t-w+1)} & \dots & z^{2(t-1)} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \dots & z^{\lceil \frac{n}{2} \rceil (t-w-1)} & z^{\lceil \frac{n}{2} \rceil (t-w)} & z^{\lceil \frac{n}{2} \rceil (t-w+1)} & \dots & z^{\lceil \frac{n}{2} \rceil (t-1)} \\ \hline z^{t-w-1} & \dots & 1 & z^{t-1+1} & z^{t-2} & \dots & z^{t-w} \\ z^{2(t-w-1)} & \dots & 1 & z^{2(t-1)+1} & z^{2(t-2)} & \dots & z^{2(t-w)} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ z^{\lfloor \frac{n}{2} \rfloor (t-w-1)} & \dots & 1 & z^{\lfloor \frac{n}{2} \rfloor (t-1)+1} & z^{\lfloor \frac{n}{2} \rfloor (t-2)} & \dots & z^{\lfloor \frac{n}{2} \rfloor (t-w)} \end{array}\right) \tag{7}$$

In this matrix, the first $\lceil \frac{n}{2} \rceil$ rows constitute a Vandermonde matrix and the last $\lfloor \frac{n}{2} \rfloor$ rows are obtained by using the upper half of G.

In another representation, we can define G as four submatrices as (4).

$$G = \begin{pmatrix} V\left([z, \dots, z^{\lceil \frac{n}{2} \rceil}], [0, \dots, t-w-1]\right) & V\left([z, \dots, z^{\lceil \frac{n}{2} \rceil}], [t-w, \dots, t-1]\right) \\ V\left([z, \dots, z^{\lfloor \frac{n}{2} \rfloor}], [t-w-1, \dots, 0]\right) & V\left([z, \dots, z^{\lfloor \frac{n}{2} \rfloor}], [t, t-2, \dots, t-w]\right) \end{pmatrix} \tag{8}$$

Where $V(x, b)$ is an $r \times c$ matrix, $x = (x_1, x_2, \dots, x_r)$ is an r-tuple of indeterminates and $b = (b_1, b_2, \dots, b_c)$ is an c-tuple of non-negative integers. The elements of matrix are obtained as $V(x, b) = [x_i^{b_j}]$.

As can be seen $max_{i,j}\{ g_{i,j}\} = z^{\lfloor \frac{n}{2} \rfloor (t-1)+1}$ which is approximately $1/2$ of that of Vandermonde matrix, this reduces the storage overhead by half.

4.1. Sharing Phase:

Consider a secret K with wL bits. First, K is divided into w segments $K_i, i = 1,2, \dots, w$, in which K_i is L bits long. Then, each segment is considered as the coefficients of a polynomial of order at most $L-1$. Next, the following steps are performed to generate n shares:

1. Generator matrix G with dimension $n \times t$ is produced as in (7).
2. $(t-w)$ random strings, $R_i, i = 1,2, \dots, t-w$ with L' bits are generated. Each string R_i is represented by the coefficients of a polynomial:

$$R_i(z) = R_{i,0} + R_{i,1}(z) + \cdots + R_{i,L'-1}\, z^{L'-1} \tag{9}$$

3. Each share $S_i, i = 1,2, \dots, n$ is calculated as follows:

$$S_i(z) = \begin{cases} \displaystyle\sum_{r=1}^{t-w} \left(R_r(z) z^{(i-1)(r-1)}\right) + \sum_{m=1}^{w} \left(K_m(z) z^{(i-1)(t-w+m-1)}\right) \bmod z^{L'} & \text{if } i < \lfloor \frac{n}{2} \rfloor \\ \displaystyle\sum_{r=1}^{t-w} \left(R_r(z) z^{(i-1)(t-w-r)}\right) + K_1(z) z^{i(t-1)+1} + \sum_{m=2}^{w} \left(K_m(z) z^{(i-1)(m-1)}\right) \bmod z^{L'} & \text{otherwise} \end{cases} \tag{10}$$

where $mod\ z^{L'}$ denotes the truncation at degree L'.

Alternatively, the encoding can be illustrated by the following notation. At first, multiplication of a matrix-vector is calculated by:

$$S'(z) = G \times \begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_{t-w} \\ K_1 \\ K_2 \\ \vdots \\ K_w \end{pmatrix} \tag{11}$$

Then, each share S_i is obtained by truncating $S_i{}'(z)$ at degree L'.

4.2. Recovery Phase:

With having t shares $S_{i_1}, S_{i_2}, \dots, S_{i_t}$ the secret can accurately be recovered. Let set $I = \{i_1, i_2, \dots, i_t\}$ contain indices of the available shares. The recovery steps can be performed by zigzag decoding algorithm explained in Section III-B.

5. Security Analysis

In this section we prove that the decodability and secrecy requirements are guaranteed in our proposed scheme. Theorem 2 shows that we can recover the secret by t shares, and additionally, by $(t-w)$ or fewer shares no information of the secret is gained. Theorem 4 provides proof for zigzag decodability. For the sake of simplicity, we make use of some lemmas as well.

Lemma 1. Any $t \times t$ square submatrix of the matrix G (defined in Section IV) is invertible, or equivalently, any t rows of G are linearly independent.

Proof: The proof of this lemma can be found in the Appendix section of the paper.

Theorem 2. Let V_A denotes the set of shares corresponding to a random subset of participants A. So, we have:

$$H(K|V_A) = \begin{cases} H(K) & m \leq t-w \\ 0 & m \geq t \end{cases} \tag{12}$$

Where $m = |A|$, i.e. number of participants of A.

Proof: Let $A = \{P_{t_0}, P_{t_1}, \dots, P_{t_{m-1}}\}$. The generator matrix G=[U V] is defined such that we have:

$$S' = G \times \begin{bmatrix} R \\ K \end{bmatrix} = (\mathrm{U} \times \mathrm{R}) \oplus (\mathrm{V} \times \mathrm{K}) = [\mathrm{S}'_1, \mathrm{S}'_2, \ldots, \mathrm{S}'_\mathrm{n}]^\mathrm{T} \quad (13)$$

where $R = [R_1, R_2, \ldots, R_{t-w}]^T$ and $K = [K_1, K_2, \ldots, K_w]^T$. According to Step 3 of Sharing phase, each share S_i is directly obtained by truncation of S'_i.

Suppose that R is selected uniformly and that K and R are mutually independent. According to Lemma 1, any t rows of G are linearly independent. Also, any $(t-w)$ rows of U and any w rows of V are linearly independent, i.e. for $m \leq (t-w)$: $rank(G) = rank(U) = m$. Hence all elements obtained by $U \times R$ are random and mutually independent. Then, we suppose that S' is a certain subset of the shares that can be obtained with uniform probability from any chosen $V \times K$. Therefore, K is independent of S' and $H(K|V_A) = H(K)$ is satisfied if $m \leq (t-w)$. This means that no information about K can be extracted.

If $m \geq t$, then $rank(G) = t$. Therefore, solving the system of linear equation 13 specifies the elements of R and K uniquely. This means that any t shares are able to recover the secret.

In the following, we prove that the zigzag decoding algorithm can be applied on shares produced by the proposed scheme. Since the proposed generator matrix does not satisfy the sufficient condition for zigzag decodability (i.e. increasing difference property), a new proof is required.

In [22], new conditions are presented for ZD codes which can reduce storage overhead provided that the matrix is selected correctly. The following results give necessary conditions for zigzag decodablity.

First for $p \in \{1,2,\ldots,n\}$ and $m, r \in \{1,2,\ldots t\}$, define $\Delta^p_{m,r} = g_{pm} - g_{pr}$.

Lemma 3. Assume that G is a generator matrix of a ZD code such that for any row indices i and j, and for any column indices m and r of G, where $i \neq j$ and $m \neq r$, we have:

1) $\Delta^i_{m,r} \neq 0$;

2) $\Delta^i_{m,r} \neq \Delta^j_{m,r}$;

3) If $g_{im} > g_{jm}$ and $\Delta^i_{m,r} > 0$, then $\Delta^i_{m,r} > \Delta^j_{m,r}$.

Then, the original message can be reconstructed by the zigzag decoding algorithm.

Proof: This lemma is proved in Theorems 1 and 2 of [22].

Theorem 4. Assume that $\mathrm{t} > 2$ zigzag decoding algorithm can be applied in our $(\mathrm{t}, \mathrm{w}, \mathrm{n})$-RSS scheme, i.e. the existence of a share (encoded packet) containing an exposed bit is always guaranteed.

Proof: For better observation, we write the difference value between consecutive components of matrix G (defined by (7)) as follows:

$$d = \left(\begin{array}{ccc|cccc} 1 & \cdots & 1 & 1 & 1 & 1 & \cdots & 1 \\ 2 & \cdots & 2 & 2 & 2 & 2 & \cdots & 2 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \lceil \frac{n}{2} \rceil & \cdots & \lceil \frac{n}{2} \rceil & \lceil \frac{n}{2} \rceil & \lceil \frac{n}{2} \rceil & \lceil \frac{n}{2} \rceil & \cdots & \lceil \frac{n}{2} \rceil \\ \hline -1 & \cdots & -1 & t-1+1 & -2 & -1 & \cdots & -1 \\ -2 & \cdots & -2 & 2(t-1)+1 & -3 & -2 & \cdots & -2 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -\lfloor \frac{n}{2} \rfloor & \cdots & -\lfloor \frac{n}{2} \rfloor & \lfloor \frac{n}{2} \rfloor (t-1)+1 & -\lfloor \frac{n}{2} \rfloor - 1 & -\lfloor \frac{n}{2} \rfloor & \cdots & -\lfloor \frac{n}{2} \rfloor \end{array}\right) \quad (14)$$

Note that $d_{i,m} = \Delta^i_{m,(m+1)}$.

It can be easily seen that the first condition of Lemma 3 holds.

Now we check condition 3. There are four cases for indices of rows i and j. In the two first cases, rows i and j both are in the upper half or the lower half G. In these cases, condition 3 is satisfied.

For the third case, consider i-th row in the upper half of matrix G and j-th row in the lower half. Without loss of generality consider $m = t - w - 1$ and $r < m$ (i.e. the left part of the matrix). For these values of m, r, we have $g_{im} > g_{jm}$, $\Delta^i_{m,r} > 0$ and $\Delta^j_{m,r} < 0$ and therefore $\Delta^i_{m,r} > \Delta^j_{m,r}$. Similarly, for $m = t - 1$ and $r < m$ (i.e. the right part of the matrix), we have $g_{im} > g_{jm}$, $\Delta^i_{m,r} > 0$ and $\Delta^j_{m,r} < 0$ and therefore $\Delta^i_{m,r} > \Delta^j_{m,r}$.

For the last case, i.e. i-th row in the lower half of matrix G and j-th row in the upper half, it can be easily seen that the condition holds. As for condition 2, in matrix d, rows i, j $(i \neq j)$ in upper half of G have different differences $\Delta^i_{m,r} = i(r-m)$ and $\Delta^j_{m,r} = j(r-m)$.

Similarly, we see this for rows i, j in lower half of G. Another case is that i, j are in different parts, for example i in upper half of matrix G and j in the lower half. If m, r both are in the left part or the right part, $\Delta^i_{m,r} > 0$ and $\backslash Delta^j_{m,r} < 0$ and therefore the condition holds. However, condition 2 may not hold when i, j are in different parts (the top and down part of G) and m, r are in different parts (the left and right part of G). But, we show that it cannot prevent the progress of ZD algorithm.

Consider we have two rows i, j and two columns m, r with $\Delta^i_{m,r} = \Delta^j_{m,r}$. Without loss of generality, consider i, j in the upper and lower half of G and $m = 1, r = t - w + 1$ (in the left and the right part of G). According to the sharing algorithm, R_1 and K_1 are multiplied by 1-th and $(t-w+1)$-th column of G, respectively. Now, suppose that the zigzag decoding algorithm runs on t shares including S_i, S_j. Since $\Delta^i_{m,r} = \Delta^j_{m,r}$, both i-th and j-th shares include $R_1 \oplus K_1$. It means that bits of R_1 and K_1 can not decode only by S_i and S_j and should use another row p where $\Delta^p_{m,r} \neq \Delta^{i\ or\ j}_{m,r}$. In the following, we show that all other rows have different differences from i, j.

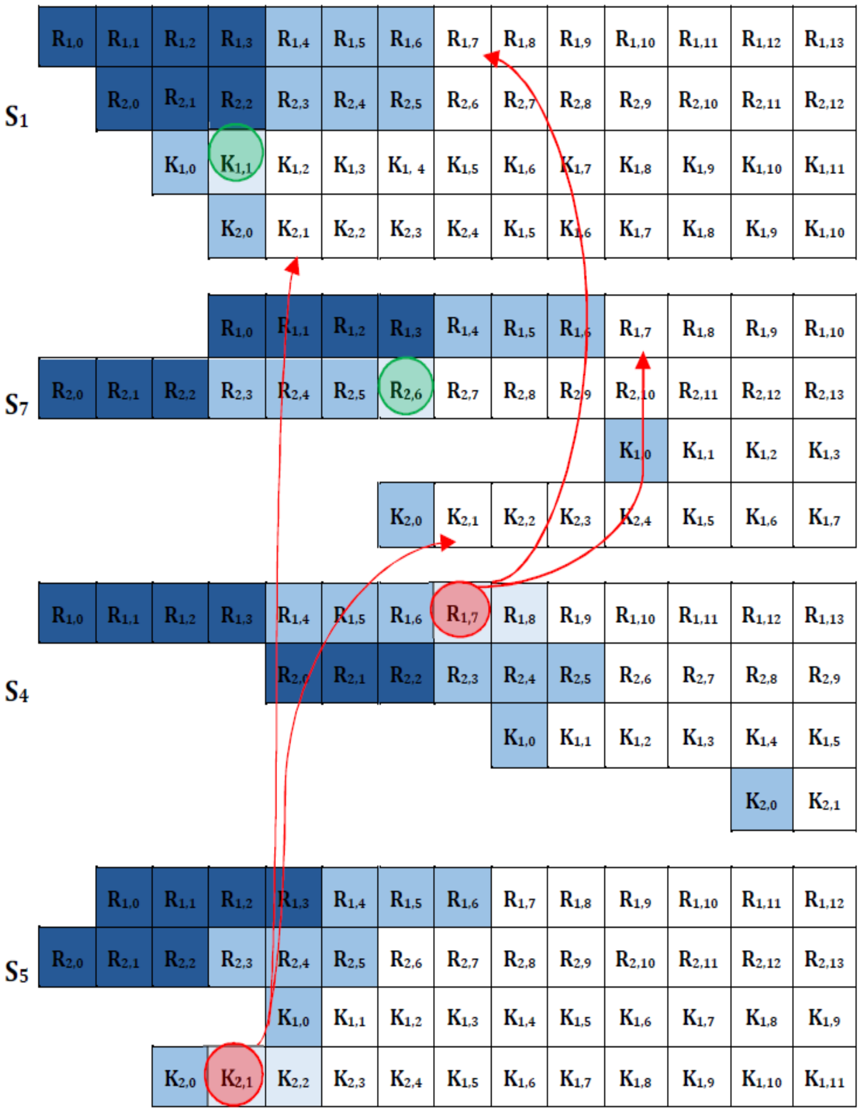


Figure 2. Recovering the secret from the proposed (4, 2, 6)-RSS scheme via zigzag decoding algorithm

Table 1. Comparison of some Boolean-based SS and RSS schemes.

Schemes	No. of computations		Support for all n , t	Only Boolean operation	Storage overhead
	Sharing phase	Recovery phase			
Kurihara et al.	tn_pL	$tn_pL + t^3n_p^3$	Yes	Yes	$l < w(n_p - 1)$
Wang-Desmedt	tnL	nL^3	Yes	No	$l \leq (n-1)$
Chen et al.	tnL	tnL	No	No	No
Deshmukh et al.	$(2^h - 2)L$	$(2^{h-1})L$	No	Yes	No
Chattopadhyay et al.	$(n)hash + 2nL$	$(n)hash + 2nL$	No	No	No
Shima-Doi	tn_pL	$tn_pL + t^3n_p^3$	Yes	Yes	$l < w(n_p - 1)$
VZD-RSS	$tn(L + tn)$	$t^2n(L + tn)$	Yes	Yes	$(n-1)(t-1)$
Our scheme	$tn(L + \frac{tn}{2})$	$t^2n(L + \frac{tn}{2})$	Yes	Yes	$\frac{n(t-1)}{2}$

All rows of upper half of G have distinct differences, then no $p(\neq i)$ has the same difference as $\Delta_{m,r}^{i}$. Similarly, all rows of lower half of G have distinct differences, then no $p(\neq j)$ has the same difference as $\Delta_{m,r}^{j}$. Therefore, for any row $p(\neq i,j)$, we have $\Delta_{m,r}^{p} \neq \Delta_{m,r}^{i\ or\ j}$.

Example. Consider (4,2,6)-RSS scheme. The generator matrix is:

$$G = \begin{pmatrix} 1 & z & z^2 & z^3 \\ 1 & z^2 & z^4 & z^6 \\ 1 & z^3 & z^6 & z^9 \\ 1 & z^4 & z^8 & z^{12} \\ z & 1 & z^4 & z^2 \\ z^2 & 1 & z^7 & z^4 \\ z^3 & 1 & z^{10} & z^6 \\ z^4 & 1 & z^{13} & z^8 \end{pmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{matrix} \tag{15}$$

Suppose that the shares corresponding to rows 1,4,5,7, are provided. In Figure 2, we show how the zigzag decoding algorithm runs.

As can be seen in (15), $\Delta_{1,4}^{1} = \Delta_{1,4}^{7}$, i.e. the difference between elements of columns 1 and 4 in rows 1 and 7 are equal. According to the proof of Theorem 4, bits of R_1 and K_2, multiplied by columns 1 and 4, should be decoded by other available shares, i.e. S_4 and S_5.

Also, $\Delta_{2,3}^{4} = \Delta_{2,3}^{5}$ and bits of R_2 and K_1 should be encoded by shares S_1 and S_7. In Figure 2, we draw a circle around the exposed bits in the current round.

6. Comparison

In this section, we compare our scheme with some similar schemes proposed by Kurihara et al. [6], Wang-Desmedt [9], Chen et al. [10], Deshmukh et al. [12], Chattopadhyay et al. [20], Shima-Doi [8], and Gong et al. [25] denoted as VZD-RSS and summarize the results in Table 1. The comparison is based on computational efficiency as well as whether there exist some limitations on the values of n and t that the schemes support. The results indicate that our scheme is the only scheme that has improved computational complexity in both sharing and recovery phases, while it has no limitation on n, t. Also, its storage overhead is almost half of the VZD-RSS.

Based on Table 1, in the sharing phase, computational complexity of our scheme is superior to VZD-RSS. In Kurihara et al.'s scheme, number of computations is tn_pL, where $n_p \geq n$ is a prime number. In the best case, $n_p = n$, but there are cases that n_p is much larger than n and therefore our scheme is more efficient than Kurihara et al.'s scheme. Chen et al.'s scheme has low computational complexity but it has a limitation and can be used only for $n = t, t+1$. Shima-Doi also achieves the same computational complexity as Kurihara et al. Deshmukh et al.'s scheme has efficient computation complexity, but it is limited to the special case of (n, n) where $n = 2^{h-1}$. In Chattopadhyay et al.'s scheme, in addition to XOR operation, it is necessary to calculate the hash function n times which imposes high computational overhead on the system. For large secret, i.e. when L grows faster than tn, the complexity of ZD-based schemes reduces to $O(tnL)$.

In the recovery phase, computational complexity of the proposed scheme is $t^2\,(L + tn/2)$ and therefore has higher efficiency than other schemes. For large secret, i.e. with increasing L, complexity of Wang and Desmedt's scheme (nL^3) has the worst efficiency compared to all other schemes. Also, computational complexity of the ZD-based schemes are $O(t^2L)$ and outperform the remaining three schemes with complexity $O(tnL)$.

We now discuss storage overhead of the compared schemes. Kurihara et al., Deshmukh et al., Chattopadhyay et al. and Wang-Desmedt's schemes do not increase the share size during sharing phase, however they may pad some bits to the secret before running this phase. That is, ultimately, their generated shares have some overhead compared to the original secret. In Kurihara et al., the secret is padded if its length is not a multiple of $w(n_p - 1)$. Wang-Desmedt's scheme pads the secret to a string of length n, which it is negligible.

The proposed scheme and VZD-RSS have also storage overhead. In VZD-RSS, the size of each share is $L + (n-1)(t-1)$. This means that the size of the overhead is $(n-1)(t-1)$. While, the proposed scheme generates the shares with the length of $L + (n)(t-1)/2$ which reduces the overhead almost by half.

Another advantage of our method is that sometimes more than one bit is exposed in each iteration of the zigzag decoding algorithm. But in the VZD-RSS method, exactly one bit is exposed in each round. This can increase speed of decoding.

7. Conclusion

This paper presents a (t, w, n)-ramp secret sharing scheme based on ZD codes. The secret recovery phase as well as sharing phase are done using only Boolean operations. Storage overhead of the proposed scheme is almost half of the overhead in existing literature. We prove that while the overhead is decreased, the scheme preserves its security. We further prove that the proposed algorithm achieves zigzag decodability, i.e. the recovery phase involves only the shift and XOR operations.

8. Appendix

This section is devoted to the proof of Lemma 1. We prove the lemma in three steps:

1. We show that it is possible to split G into two submatrices whose rows are linearly independent.
2. We show that if we replace any rows of one of these submatrices with a row of the {other submatrix}, then the rows of the resulting matrix are still linearly independent.
3. We show that the claim stated in step II is valid for any number of rows.

Step I. For simplicity, we consider $n/2 = t$. First, we partition G into two non-overlapping submatrices A and B as follows.

$$G = \left(\begin{array}{cccccc} 1 & z & \dots & z^{t-w-1} & \dots & z^{t-1} \\ 1 & z^2 & \dots & z^{2(t-w-1)} & \dots & z^{2(t-1)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & z^t & \dots & z^{t(t-w-1)} & \dots & z^{t(t-1)} \\ \hline z^{t-w-1} & z^{t-w-2} & \dots & 1 & \dots & z^{t-w} \\ z^{2(t-w-1)} & z^{2(t-w-2)} & \dots & 1 & \dots & z^{2(t-w)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ z^{t(t-w-1)} & z^{t(t-w-2)} & \dots & 1 & \dots & z^{t(t-w)} \end{array}\right) = \begin{pmatrix} A \\ B \end{pmatrix}$$

To prove that the rows of these submatrices are linearly independent, we show that A and B are invertible.

1. A is a Vandermonde matrix which is known to be invertible iff the values of its second column are non-repetitive [29].
2. The submatrix B is a special columnar permuted form of Vandermonde matrix. We know that the columnar elementary operations do not change the rank of a matrix, accordingly, B is invertible [30].

Step II. We first show that if we replace an arbitrary row of A with any row of B, the resulting matrix has t independent rows. To do so, we show that the new row is independent of other remaining rows of A.

To prove this, we make A upper triangular and call it A' as (16).

$$A' = \begin{pmatrix} 1 & z & \dots & z^{i-2} & z^{i-1} & \dots & z^{t-1} \\ 0 & z^2 - z & \dots & z^{2(i-2)} - z^{i-2} & z^{2(i-1)} - z^{i-1} & \dots & z^{2(t-1)} - z^{t-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & z^{i(i-1)} - z^{i-1} & \dots & z^{i(t-1)} - z^{t-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & z^{t(t-1)} \end{pmatrix} \quad (16)$$

Afterwards, consider the matrix A'' obtained by replacing the i-th row of A' with the j-th row of B as (17).

$$A'' = \left(\begin{array}{ccccccc} 1 & z & \dots & z^{i-2} & z^{i-1} & \dots & z^{t-1} \\ 0 & z^2 - z & \dots & z^{2(i-2)} - z^{i-2} & z^{2(i-1)} - z^{i-1} & \dots & z^{2(t-1)} - z^{t-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \hline z^{j(t-w-1)} & z^{j(t-w-2)} & \dots & & & \dots & z^{j(t-w)} \\ \hline \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & z^{t(t-1)} \end{array}\right) \quad (17)$$

Note that since we choose the coefficients based on the value of $(i-1)$-th cell of i-th row to make it 0, $A''[i][i]$ will never be zero.

We can convert the first $(i-1)$ values of the i-th row to 0 by the leading coefficient of its previous rows. But we cannot make the i-th value 0 at the same time, since there is no row in which its i-th value is the leading coefficient, and the sum of previous rows cannot make $(i-1)$-th and i-th value 0, simultaneously, in the light of the fact that there is no repetitive value in none of the G's columns. So, at least we will have $z^{j(t-i-1)+1}$ left in the new i-th term.

Step III. Finally, giving an incremental construction algorithm, we explain that we can generalize the result of previous step for any number ($\leq t$) of rows. Instead of replacing all new rows simultaneously, we are able to do that one by one. Using the same argument as step II, after altering one of the rows we have again a submatrix in its previous state, meaning that it has still t linearly independent rows and it is still upper-triangularizable such that all of its diagonal elements are non-zero. So, for the second row, we can continue as in the first one and so on.

This completes the proof, by virtue of the fact that the same happens when replacing a row of B with a row of A.

9. References

[1] Shamir, A., "How to share a secret", *Commun. ACM*, Vol. 22, No. 11, pp. 612–613, 1979.

[2] Hineman, A., and Mario, B., "A modified Shamir secret sharing scheme with efficient encoding", IEEE Communications Letters, Vol. 26.4, pp. 758-762, 2022.

[3] Shiina, N., "How to convert 1-out-of-n proof into k-out-of-n proof", Proc SCIS2004, pp. 1435–1440, 2004.

[4] Kurihara, J., Kiyomoto, S., Fukushima, K., and Tanaka, T., "A fast (3, n)-threshold secret sharing scheme using exclusive-or operations", *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, Vol. 91, No. 1, pp. 127–138, 2008.

[5] Kurihara, J., Kiyomoto, S., Fukushima, K., and Tanaka, T., "A new (k, n)-threshold secret sharing scheme and its extension", *in International Conference on Information Security*, pp. 455–470, Springer, 2008.

[6] Kurihara, J., Kiyomoto, S., Fukushima, K., and Tanaka, T., "A fast (k, L, n)-threshold ramp secret sharing scheme", *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, Vol. 92, No. 8, pp. 1808–1821, 2009.

[7] Beimel, A., and Othman, H., "Evolving ramp secret sharing with a small gap", EUROCRYPT 2020, 2020.

[8] Shima, K., and Doi, H., "New Proof Techniques Using the Properties of Circulant Matrices for XOR-based (k, n) Threshold Secret Sharing Schemes", *J. Inf. Process.*, Vol. 29, pp. 266–274, 2021.

[9] Wang, Y., and Desmedt, Y., "Efficient secret sharing schemes achieving optimal information rate", *in 2014 IEEE Information Theory Workshop (ITW 2014)*, IEEE, pp. 516–520, 2014.

[10] Chen, L., Laing, T. M., and Martin, K. M., "Efficient, XOR-Based, Ideal (t, n)- threshold Schemes", *in International Conference on Cryptology and Network Security*, pp. 467–483, Springer, 2016.

[11] Shima, K., and Doi, H., "A hierarchical secret sharing scheme over finite fields of characteristic 2", *J. Inf. Process.*, Vol. 25, pp. 875–883, 2017.

[12] Deshmukh, M., Maroti, Neeta, N., and Mushtaq, A., "Secret sharing scheme based on binary trees and Boolean operation", *Knowledge and Information Systems*, Vol. 60, No. 3, pp. 1377-1396, 2019.

[13] Pande, D., Rawat, A. S., Deshmukh, M., and Singh, M., "Single secret sharing scheme using chinese remainder theorem, modified Shamir's scheme and XOR operation", *Wireless Personal Communications*, Vol. 130, No. 2, pp. 957-985, 2023.

[14] Kabirirad, S., and Eslami, Z., "Improvement of (n, n)-multi-secret image sharing schemes based on Boolean operations", *J. Inf. Secur. Appl.*, Vol. 47, pp. 16–27, 2019.
[15] Bisht, K., and Deshmukh, M., "A novel approach for multilevel multi-secret image sharing scheme", *J. Supercomput.*, pp. 1–35, 2021.
[16] Paul, A., Kandar, S., and Dhara, B. C., "Boolean operation based lossless threshold secret image sharing", *Multimedia Tools and Applications*, Vol. 81 No. 24, pp. 35293-35316, 2022.
[17] Huang, P.-C., Chang, C.-C., Li, Y.-H., and Liu, Y., "Enhanced (n, n)-threshold QR code secret sharing scheme based on error correction mechanism", *J. Inf. Secur. Appl.*, Vol. 58, pp. 102719, 2021.
[18] Kabirirad, S., and Eslami, Z., "A (t, n)-multi secret image sharing scheme based on Boolean operations", *J. Vis. Commun. Image Represent.*, Vol. 57, pp. 39–47, 2018.
[19] Nag, A., Singh, J. P., and Singh, A. K., "An efficient Boolean based multi-secret image sharing scheme", *Multimed. Tools Appl.*, pp. 1–25, 2019.
[20] Chattopadhyay, A. K., Nag, A., Singh, J.P., and Singh, A. K., "A verifiable multi-secret image sharing scheme using XOR operation and hash function", *Multimedia Tools and Applications*, Vol 80, pp. 35051-35080, 2021.
[21] Sung, C. W., and Gong, X., "A ZigZag-decodable code with the MDS property for distributed storage systems", *in 2013 IEEE International Symposium on Information Theory, IEEE,* pp. 341–345, 2013.
[22] Dai, M., Sung, C. W., Wang, H., Gong, X., and Lu, Z., "A new zigzag-decodable code with efficient repair in wireless distributed storage", *IEEE Trans. Mob. Comput.*, Vol. 16, No. 5, pp. 1218–1230, 2016.
[23] Hou, H., Lee, P. P., and Han, Y. S., "ZigZag-decodable reconstruction codes with asymptotically optimal repair for all nodes", *IEEE Trans. Commun.,* Vol. 68, No. 10, pp. 5999–6011, 2020.
[24] Lu, S., Zhang, C., and Dai, M., "CP-BZD Repair Codes Design for Distributed Edge Computing", *in 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS), IEEE,* pp. 722–727, 2020.
[25] Gong, X., Hu, P., Shum, K. W., and Sung, C. W., "A zigzag-decodable ramp secret sharing scheme", *IEEE Trans. Inf. Forensics Secur.*, Vol. 13, No. 8, pp. 1906–1916, 2018.
[26] Gong, X., and Sung, C. W., "Zigzag decodable codes: Linear-time erasure codes with applications to data storage", *J. Comput. Syst. Sci.*, Vol. 89, pp. 190–208, 2017.
[27] Iwamoto, M., and Yamamoto, H., "Strongly secure ramp secret sharing schemes for general access structures", *Inf. Process. Lett.*, Vol. 97, No. 2, pp. 52–57, 2006.
[28] Gollakota, S., and Katabi, D., "Zigzag decoding: Combating hidden terminals in wireless networks", *in Proceedings of the ACM SIGCOMM 2008 conference on Data communication,* pp. 159–170, 2008.
[29] Trappe, W., "Introduction to cryptography with coding theory", Pearson Education India, 2006.
[30] Hoffman, K., and Kunze, R., "Linear Algebra, Prentice-Hall", *Inc Englewood Cliffs N. J.*, pp. 122–125, 1971.